(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 833 517 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
01.04.1998 Bulletin 1998/14

(51) Int Cl.[6]: **H04N 7/30**

(21) Application number: **97307141.8**

(22) Date of filing: **15.09.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **25.09.1996 US 27436 P**
07.03.1997 US 813218

(71) Applicant: **AT&T Corp.**
New York, NY 10013-2412 (US)

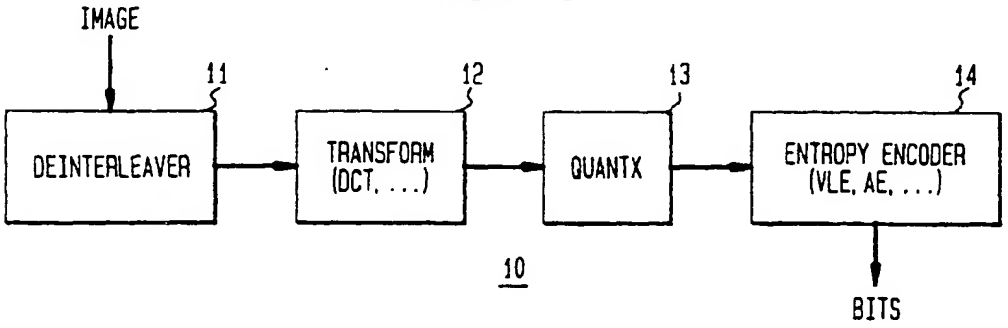
(72) Inventors:
- **Puri, Atul**
  Riverdale, New York 10463 (US)
- **Wus, John**
  Warminster, Pennsylvania (US)
- **Schmidt, Robert Louis**
  Howell, New Jersey 07731 (US)
- **Li, Weiping**
  Bethlehem, PA 18017 (US)

(74) Representative: **Pearce, Anthony Richmond**
MARKS & CLERK,
Alpha Tower,
Suffolk Street Queensway
Birmingham B1 1TT (GB)

(54) Fixed or adaptive deinterleaved transform coding for image coding and intra coding of video

(57) A coding strategy efficiently codes intra (macroblocks, regions, pictures, VOP) data. This strategy uses two basis approaches, a Fixed Deinterleaved Transform Coding approach, and an Adaptive Deinterleaved Transform Coding approach. Furthermore, within each approach, two types of coders are developed. One coder operates on an entire picture or VOPs and the other coder operates on small local regions. Using coders and decoders of the present invention, efficient coding at a range of complexities becomes possible, allowing suitable tradeoffs for a variety of applications.

FIG. 1



EP 0 833 517 A2

Description

BACKGROUND OF THE INVENTION

This patent applicaiton is based on provisional application 60/027,436 filed on September 25, 1996.

The present invention relates generally to methods and apparatuses for coding images and intra coding of video and more particularly to a method and apparatus for coding images and intra coding of video using transform coding. Intra coding is important for applications requiring simple encoding/decoding, low delay, a high degree of error robustness or a high level of interactivity. Examples of such applications include image/video on the Internet, wireless video, networked video games, etc.

The state of the art standards in image coding and intra coding of video employ transform coding (e.g., Discrete Cosine Transformation, DCT), which involves partitioning an image into non-overlapping blocks of 8x8 size and coding in units of a 2x2 array of luminance (Y) blocks and a corresponding chrominance block of Cr and a block of Cb signal (together referred to as a macroblock). Improvements in performance of intra coding have been obtained by predicting the DC coefficient of DCT blocks by using previously reconstructed DC coefficients. Recently, further improvements have been obtained in MPEG-4 by predicting AC coefficients as well.

Over the past several years, many researchers have followed a different approach that uses wavelets instead of transform coding. These researchers have reported substantial improvements, at the expense of increased complexity. Recently, variations of wavelet coding that use subsampling prior to coding have emerged (some of which are currently being experimented with in MPEG-4), that provide an even higher performance by using advanced quantization techniques that take advantage of subsampling, however, the complexity of such schemes is extremely high.

The present invention is therefore directed to the problem of developing a method and apparatus for coding images and video that has a high coding efficiency yet relatively low coding complexity.

SUMMARY OF THE INVENTION

The present invention solves this problem by using a deinterleaving step prior to the transformation step in combination with a suitable quantization technique. Deinterleaving is a more flexible form of subsampling. The method and apparatus of the present invention thus achieve the coding efficiency of wavelet coding, at the expense of only a small improvement in complexity over traditional transform coding but at a significantly reduced complexity relative to wavelet coding.

The present invention includes two forms of deinterleaving in the encoding process. According to the present invention, improved intra coding is achieved by fixed or adaptive deinterleaving, transform (e.g., DCT), quantization with extensions and entropy encoding. (e.g., Variable Length Encoding, VLE, or Arithmetic Encoding, AE). According to the present invention, there are two main approaches to coding. The first uses fixed deinterleaving, and the second uses adaptive deinterleaving. It now becomes possible as a result of the present invention to use a simple transform coding method that achieves high coding efficiency using a fixed deinterleaving approach, or to use a slightly more complex method that produces somewhat better results based on an adaptive deinterleaving approach, and yet both of these approaches are less complex than wavelet coding and achieving the same or nearly the same coding efficiency.

According to advantageous implementations of the present invention, for each approach, two variations exist, one intended for separate motion/texture part of the MPEG-4 Verification Model (VM) (global transform) coding and the other intended for combined motion/texture part of the VM (local transform) coding.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG 1 depicts the basic encoder block diagram for Global Deinterleaved Transform (GDT) coding according to the present invention.

FIG 2 shows the corresponding decoder for the encoder shown in FIG 1 according to the present invention.

FIG 3 depicts the basic encoder block diagram of Local Deinterleaved Transform (LDT) coding according to the present invention.

FIG 4 shows the corresponding decoder for the encoder shown in FIG 3 according to the present invention.

FIG 5 shows a simple example of deinterleaving a region by a factor of 2 both in the horizontal and vertical direction to generate subregions.

FIG 6 shows an 8x8 array of subpictures, each 22x18 in size that results from application of 8:1 deinterleaving in horizontal and vertical directions to the luminance signal in GDT coding for _____ (QCIF) resolution.

FIG 7 shows a 4x4 array of subpictures each 8x8 in size that result from application of 4:1 deinterleaving in horizontal and vertical directions to 32x32 regions of luminance signal in LDT coding for QCIF resolution.

FIG 8 depicts one method of extended quantization, QuantX Method 1, according to the present invention.

FIG 9 shows the inverse operation of the extended quantization method shown in FIG 8, QuantX Method 1, according to the present invention.

FIG depicts an example of quantized DC coefficients prediction used in the present invention.

FIG 11 shows an example of AC coefficient prediction structure employed in the present invention.

FIG 12 depicts another method of extended quantization, QuantX Method 2, according to the present invention.

FIG 13 shows the inverse operation of the extended quantization method shown in FIG 12, QuantX Method 2, according to the present invention.

FIG 14 depicts another method of extended quantization, QuantX Method 3, according to the present invention.

FIG 15 shows the inverse operation of the extended quantization method shown in FIG 14, QuantX Method 3, according to the present invention.

FIG 16 shows the block diagram of an Adaptive Global Deinterleaved Transform (AGDT) encoder 150 used in the present invention.

FIG 17 shows the block diagram of the AGDT decoder used in the present invention, which corresponds to the AGDT encoder shown in FIG 16.

FIG 18 shows a block diagram of an Adaptive Local Deinterleaved Transform (ALDT) encoder used in the present invention.

FIG 19 shows the ALDT decoder used in the present invention, which corresponds to the encoder of FIG 17.

FIG 20 shows an example of quadtree segmentation employed in the present invention.

DETAILED DESCRIPTION

According to the present invention, a significant improvement in efficiency of intra coding within the framework of transform coding is now possible. The present invention has been designed for use with MPEG-4. Often in MPEG-1/2 video coding, intra coding is employed for pictures coded by themselves, which are called I-pictures, or for intra macroblocks in predictively coded pictures (P-pictures or B-pictures). Besides pictures or macroblocks, MPEG-4 also introduces the concept of Video Objects (VO) and Video Object Planes (VOPs).

In MPEG-4 coding a scene can be partitioned into a number of Video Objects each of which can be coded independently. A VOP is a snapshot in time of a Video Object. In fact, a picture then becomes a special case of a VOP that is rectangular in shape.

MPEG-4 coding also includes coding of VOPs of different types, such as I-VOPs, P-VOPs, and B-VOPs, which are generalizations of I-pictures, P-pictures, and B-pictures, respectively. Thus, in addition to coding of I-pictures, and intra macroblocks the present invention can also be used for coding of I-VOPs, both rectangular and arbitrary in shape.

The main functionality addressed by the present invention is a coding efficiency of I-VOPs, although this approach is extendable to coding of P-VOPs and B-VOPs. An additional functionality that could be indirectly derived from the present invention is spatial scalability.

The present invention achieves a significant improvement in intra coding efficiency (by a factor of 1.5 or more) and can be obtained while still employing the DCT coding framework by requiring the addition of deinterleaving and extended quantization. In general, intra coding efficiency can also be somewhat improved by further improving DC coefficient prediction, incorporating AC coefficient predictions and scanning adaptations, however, even after combining all these techniques the improvements may be relatively small as compared to the potential improvement of the present invention.

For I-VOPs coding in MPEG-4, FIG 1 depicts the basic encoder 10 block diagram of Global Deinterleaved Transform (GDT) coding according to the present invention. At the input to the deinterleaver 11, the image is in pixel format with each pixel being represented by the three components of luminance, chrominance and saturation, which are digital values. These digital values are fed into the deinterleaver 11, which separates contiguous samples within the image. In other words, the deinterleaver 11 separates the pixels sets into a number of pixel subsets, but does so by creating subsets from non-contiguous pixels. This then requires specification of the separation pattern used. Each subset contains several of the digital samples, however, the samples within a given subset were not contiguous with each other in the original image.

The transform operation 12 converts the digital pixel values within each subset into transform coefficients, such that most of the energy is packed in a few coefficients. In this step, for example, Discrete Cosine Transform (DCT) can be used. Other known transform techniques can also be used, such as _____ .

The transformer 12 receives the pixel subsets from the deinterleaver 11. Each subset contains several pixels, with each pixel being represented by the three values of chrominance, luminance and saturation, or some equivalent color system. The transformer 12 then outputs coefficients representing the spatial frequency components of the values within each subset. While there is no real compression at this point, the DCT transform groups the data that enables the latter processes to significantly reduce the data. The DCT transform defines most of the information within the subset in the lower spatial frequencies and many of the higher spatial frequencies come out to be zero, resulting in compression later on. The output of the transformer 12 then is a block of coefficients, one block for each subset created by the deinterleaving process 11.

The QuantX process 13 includes normal quantization plus some extensions to improve coding efficiency and prepares the data for entropy encoding 14. This will be described in detail below, as three different QuantX processes 13 are presented herein. The output of the QuantX process 13 is a block of bits, one for each subset created in the deinterleaving process 11.

Following the QuantX process 13 is the encoding process 14. In this case entropy encoding is used. Any form of entropy encoding will suffice. Variable Length Encoding (VLE) is one example of entropy encoding. Arithmetic Encoding (AE) is another example. The coded bitstream generated by the Entropy Encoder 14 can

now be stored or transmitted.

Other known entropy encoding can be used.

FIG 2 shows the decoder 20 corresponding to the encoder 10 shown in FIG 1. The decoder 10 for Global Deinterleaved DCT Encoding of I-VOPs includes an entropy decoder 21, an inverse quantization 22, an inverse transform 23 and a reinterleaver 24. The entropy decoder 21 inverts codewords back to coefficient data. The inverse quantization 22 performs the inverse operation of the quantization 13 plus some extensions performed in the encoder 10. The inverse transform 23 performs the inverse operation of the transform 12, and the reinterleaver 24 performs the inverse of the deinterleaver 11.

The coded bitstream is fed into the entropy decoder 21. The entropy decoder 21 outputs a block of data similar to that input to the entropy encoder 14. Due to the deinterleaving 11 performed on the coding side, this block of data is subgrouped into blocks corresponding to the subsets created by the deinterleaver 11. The entropy decoder 21 outputs the block of data to the Inverse QuantX 22.

The Inverse QuantX 22 will be described in detail below, as three different processes are presented herein, depending upon the coding process performed. The Inverse QuantX 22 feeds its output to the inverse transform 23. The output of the Inverse QuantX 22 is a block of coefficients, which can be further subgrouped according to the subsets created by the deinterleaving process 11.

The inverse transform 23 then performs the inverse transform operation on each subblock of coefficients to convert it to a subblock of pixel subsets. These subblocks are now ready for reinterleaving. The reinterleaver 24 then reconstructs the original order in which the pixels appeared.

FIG 3 depicts the basic encoder 30 block diagram of LDT coding. The main difference with respect to FIG 1 is that prior to deinterleaving an input VOP or picture is segmented into local regions. These regions can be either square (blocks) or even arbitrary in shape. With such segmentation, some of the coding details related to transform size and QuantX may change.

The image in pixel format is fed into the local regions segmenter 31, which outputs the segmented image signal to the deinterleaver 32. In this case, the local regions segmenter 31 creates subsets of pixels, which are contiguous. Then, in the deinterleaving step 32, these subsets are further partitioned so that the resulting partitioned subsets each contain non-contiguous pixels.

As in FIG 1, the remaining process is the same. The deinterleaver 32 passes its output to the transformer 33, which in turn feeds the QuantX 34, which feeds the entropy encoder 35, which outputs the coded bitstream.

FIG 4 shows the corresponding decoder 40 for the encoder 30 shown in FIG 3. The main difference with respect to the GDT decoder shown in FIG 2 is the addition of the local regions assembler 45 (e.g., block unformatter) at the end of the decoding process. The local regions assembler 45 performs the inverse operation of the local regions segmenter 31. The LDT decoder 40 includes the entropy decoder 41, the inverse quantization 42, the inverse transform 43 the reinterleaver 44, and the local regions assembler 45.

The process in FIG 4 is identical to the process in FIG 2, except that each step in the process is being performed on a partitioned subset relative to FIG 2. For example, the encoded bits are fed into the entropy decoder 41. These bits are ordered by the local regions created in the encoding process 30. Thus, each step is performed on each local region group separately. The decoder 41 then outputs groups of blocks of bits to the QuantX 42, which then creates the groups of blocks of coefficients necessary for the inverse transform process 43. The inverse transform process 43 then outputs the groups of pixels to the reinterleaver 44, which reinterleaves the pixels within each group. Output from the reinterleaver 44 is therefore the local regions created by the local region segmenter. One can view the decoding process 40 in this case as being performed on each local region separately. At the end of this decoding process 40, the local regions are then assembled by the local regions assembler 45 to reconstruct the pixel image as it was presented to the local regions segmented 31.

We shall now describe some of the steps within the process in more detail. These include deinterleaving, and QuantX.

Deinterleaving

Deinterleaving is the process of separating an input picture (or a region) into subpictures (or subregions) such that the neighboring samples in the input picture (or region) are assigned to different subpictures (or subregions). The resulting subregions or subpictures thus contain samples that were not contiguous in the original picture.

FIG 5 shows a simple example of deinterleaving a region by a factor of 2 both in the horizontal and vertical direction to generate subregions. The original picture 51 being composed of pixels (o, x, +, -) is deinterleaved into four subpictures 52-55. Every other element of the first row (o, x, o, x, o, x) is then assigned to the first row of subpicture 51 (o, o, o) and the first row of the second subpicture 52 (x, x, x). The same is true for the remaining odd rows. The even rows are assigned to the third and fourth subpictures (+, +, +) and (-, -, -), respectively, and split as before. Essentially, each pixel (p_{ij}) is assigned to the subpictur$_{k,m}$ where k=mod(i/n) and m=mod(j/n) and becomes element $p_{r,s}$ in that subpicture, where r=(i-k)/n and s=(j-m)/n.

For example, if we let where n=2, as in FIG 5, we note that element 56 (i.e., p_{23}), is assigned to subpicture 01 (element 53), that is, k=mod(2/2)=0 and m=mod(3/2)=1. If we examine subpicture 01 (element 53), we note that element 57 appears as pixel 11 in that subpicture,

and r=(i-k)/n=(2-0)/2=1 and s=(j-m)/n=(3-1)/2=1.

In this example, in GDT coding, the deinterleaving factor is fixed to 8:1 for QCIF input resolution (176x144). For this resolution, FIG 6 shows an 8x8 array, 63, of subpictures, each subpicture 62 being 22x18 in size that results from application of 8:1 deinterleaving in horizontal and vertical directions to the luminance signal. Also, each of the chrominance components are deinterleaved by a factor of 4:1 resulting in a 4x4 array of subpictures, each of size 22x18.

On the other hand, in LDT coding, the deinterleaving factor is fixed to 4:1 for QCIF input resolution. FIG 7 shows a 4x4 array, 73, of subpictures each subpicture 72 being 8x8 in size that results from application of 4:1 deinterleaving in horizontal and vertical directions to 32x32 regions of luminance signal. In this case, the chrominance components are deinterleaved by a factor of 2:1 resulting in a 2x2 array of subregions, each 8x8 in size.

DCT

Two dimensional DCT is applied to deinterleaved subpictures or subregions. In GDT coding at QCIF resolution, the size of DCT is chosen to be 22x18 both for the luminance as well as the chrominance components. In LDT coding at QCIF resolution, the size of DCT is chosen to be 8x8 both for the luminance as well as the chrominance components.

QuantX Choices

The normal scalar quantization needs to be modified to take into account the fact that the transform coding is performed on deinterleaved data. Beyond quantization, the coefficient prediction of experiment may also be more effective in increasing coding efficiency due to higher correlation between coefficients of deinterleaved adjacent subpictures (subregions). Another approach is to exploit this correlation by forming vectors of coefficients of the same spectral frequency and performing DCT coding on such vectors (blocks). Finally, yet another alternative is to use vector quantization or a specific variation called Lattice Vector Quantization (LVQ) being examined in MPEG-4. These various approaches are referred to here as QuantX and offer different tradeoffs in performance versus complexity and the right one may be selected based on the application.

QuantX Method 1: Quantization and DCT Coefficient Predictions

This method is explained by reference to FIG 8. The signal input to the QuantX 80 is received by a quantizer 81, whose output signal is split. One path feeds a DC & AC Coefficient Predictor 82, and another path feeds one input of a subtractor 83. The output of the DC & AC Coefficient Predictor 82 feeds the other input of the subtractor 83. The output of the DC & AC Coefficient Predictor 82 is subtracted from the output of the quantizer 81 and fed into a scanner 84, such as a zigzag scanner.

In GDT coding, the DCT coefficient subpictures of size 22x18 are quantized by the normal scalar quantization and then the coefficient subpictures are predicted based on previously quantized coefficient subpictures and coefficient difference subpictures are formed. In LDT coding, a very similar operation takes place on DCT coefficient subregions of size 8x8. The difference coefficients are scanned (e.g., zigzag scanned) to form (run, level) events.

FIG 9 shows the inverse operation of the QuantX shown in FIG 8. The signal input to the Inverse QuantX 90 is received by the Inverse Scanner 91. The output of the inverse scanner 91 is fed to one input of an adder 92. The second input of the adder 92 is from the output of a DC & AC Coefficient Predictor 93, which receives its input from the output of the adder 92. The output of the adder 92 is also passed to the inverse quantizer 94, which outputs the desired signal.

A scheme for quantized DC coefficients prediction is illustrated in FIG 10. In GDT coding, the DC prediction of leftmost subregion (subpicture) is selected to 128 while in LDT coding, using 1 bit overhead, selection between DC values of horizontally or vertically adjacent subregions (subpictures) is made. For the remaining subregions (subpictures) in the first row, DC prediction uses previous subregion (subpicture) DC value. For the first subregion (subpicture) of the second row, DC prediction is taken from the subregion (subpicture) above; all the other subregions (subpictures) of that row use Graham's predictor adaptively by selecting from horizontal and vertical adjacent subregion (subpicture) DC values without overhead. The prediction process for the second row is repeated for subsequent rows.

We now discuss how AC coefficient predictions are made. FIG 11 shows an example of AC coefficient prediction structure employed. In the case of LDT coding with 8x8 subregions, 2 rows and 2 columns of AC coefficient predictions for a subregion may be used. In case of larger subregion sizes in LDT coding or larger pictures in GDT coding, more AC coefficients may be predicted; the number and the structure of coefficients being predicted can be different but the basic principle of prediction remains the same.

For the left-most subregion (subpicture) of a region (picture), AC coefficient predictions are reset to 0. For the subsequent subregions (subpictures) in the first row of subregions, the L-shaped highlighted area (without DC coefficient) is predicted from subregions (subpicture). For the first subregion of second row of subregions, the same L-shaped area is predicted from the subregion immediately above. Subsequent subregions of second row are predicted by using first two columns of coefficients from previous subregion and the first two rows from the subregion above. There is an overlap of 1 coefficient (AC 11), which is resolved by averaging the

two prediction choices for this coefficient to generate a single prediction coefficient. The prediction process for the second row is repeated for subsequent rows.

Further, the potential of making the prediction process adaptive (with or without overhead) is also possible.

The difference coefficient subpictures of size 22x18 in GDT coding and subblocks of size 8x8 in LDT coding are zigzag scanned to form (run, level) events.

QuantX Method 2: Quantization and DCT of DCT Coefficient Vectors

FIG 12 illustrates the operations employed in this method of QuantX. In GDT coding of QCIF pictures, the DCT coefficient subpictures of size 22x18 are prequantized by a small quantization level (Qp= 2 or 3) to reduce their dynamic range and then vectors (of 8x8 size for luminance and 4x4 size for chrominance) are generated by collecting all coefficients of the same frequency through all of the subpictures, DCT and quantized. In LDT coding with regions of size 32x32, a very similar operation takes place resulting in coefficient vectors (of 4x4 size for luminance and 2x2 size for chrominance); these vectors are DCT and quantized. In GDT coding, quantized DCT coefficient vectors of 8x8 size for luminance and 4x4 size for chrominance are zigzag scanned to form (run, level) events. In LDT coding, quantized DCT coefficient vectors of 4x4 size for luminance and 2x2 size for chrominance are zigzag scanned for form (run, level) events.

Referring to FIG 12, the QuantX 120 includes a prequantizer 121, a vector formatter 122, a transform 123, a quantizer 124 and a scanner 125. The prequantizer 121 receives the signal input to the QuantX 120, and outputs its signal to the vector formatter 122. The vector formatter feeds its output to the transform 123, which in turn feeds the quantizer 124, which feeds the scanner 125. The scanner outputs its signal as the output of the QuantX 120.

The Inverse Operation 130 of the QuantX 120 shown in FIG 12 is shown in FIG 13. The input to the inverse QuantX 130 is fed to the inverse scan 131, which feeds the inverse quantizer 132, which in turn feeds the inverse transform 133. The vector unformatter 134 receives the output from the inverse transform 133 and outputs its signal to the inverse prequantizer 135, whose output represents the output of the inverse QuantX 130.

QuantX Method 3: Lattice Vector Quantization of DCT Coefficient Vectors

FIG 14 illustrates the operations employed by this method of QuantX. The signal input to the QuantX 140 is received by the Vector Formatter 141, which passes its output to the Dimension Reducer 142, which in turn feeds its output to the Vector Quantizer 143. The Vector Quantizer 143 then passes its output to the Vector Quantization Indices Orderer 144, whose output represents the output of the QuantX 140.

In GDT coding of QCIF pictures, using DCT coefficient subpictures of size 22x18, vectors (of 8x8 size for luminance and 4x4 size for chrominance) are generated by collecting all coefficients of the same frequency through all of the subpictures; these vectors are quantized by the LVQ. In LDT coding of region size 32x32, a very similar operation takes place resulting in coefficient vectors (of 4x4 size for luminance and 2x2 size for chrominance); these vectors are also quantized by LVQ. Since VQ often requires small blocks for manageable size codebooks (or in LVQ, a manageable complexity), a reduction of vector dimension may be necessary and is accomplished in Dimension Reducer, which can be as simple an operation as dividing a vector of coefficients into sub-vectors or something more sophisticated. The process of LVQ is not described herein and is discussed in literature. Briefly, however, first LVQ of dimension 16 is tried, if it produces errors higher than a threshold then the LVQ of dimension 4 is tried. Also after LVQ, the LVQ indices of the entire picture or region may be ordered for increased efficiency, this process takes place in VQ Indices Orderer.

The Inverse Operation 150 of the QuantX 140 shown in FIG 14 is shown in FIG 15. The input to the inverse QuantX 150 is fed to the Vector Quantization Indices Reorderer 151, which feeds the inverse vector quantizer 152, which in turn feeds the dimension normalizer 153. The vector unformatter 154 receives the output from the dimension normalizer 153 and outputs its signal as the output of the inverse QuantX 150. As in the QuantX 140, in the Inverse QuantX 150, first LVQ of 16 is tried. If it produces errors higher than a threshold, then the LVQ of dimension 4 is tried. The specification of LVQ is the same as that in experiment T. 5 in the MPEG 4 trials.

Entropy Coding

We now discuss a VL coding and decoding method for coefficient (run, level) events, which are coded exploiting statistical variations to achieve even further efficiency.

In GDT coding if extended QuantX Method 1 is employed, a maximum run of 396 is possible and a level of at least ±255 needs to be supported. For coding of luminance run/level events, the intra VLC table of U.S. Patent Application No. 08/###,###, entitled "Adaptive and Predictive Coding for Image Coding and Intra Coding of Video" by Puri, Schmidt and Haskell is employed. U.S. Patent Application No. 08/###,### is hereby incorporated by reference as if recited herein in its entirety. However, since this table supports only a maximum run of 64 and level of ±128 (same as the MPEG-4 VM) it is extended to outside of this region by appending one extra bit for level and three extra bits for run and thus uses up to 25 bits. For coding of chrominance run/level events, the VLC table used is the one in the VM extend-

ed to support a maximum run of 396 and level of ±255 by escaping outside of the currently supported region by appending one extra bit for level and three extra bits for run, thus using up to 26 bits. In case of LDT coding, since the subregion size is 8x8, the VLC table of the aforementioned earlier patent application that was incorporated by reference, is employed for luminance and the VLC table of VM is employed for chrominance; both of these tables do not require any extensions.

If extended quantization QuantX Method 2 is employed, in GDT coding, since the vector size is 8x8, the VLC table of the previously incorporated by reference patent application is employed for luminance and the VLC table of VM is employed for chrominance; both of these tables do not require any extensions. In the case of LDT coding, a maximum run for luminance is 15 and that for chrominance is 3; in this case new tables which are subsets of the previously incorporated patent application are employed.

If extended quantization QuantX Method 3 is employed, VLC tables used are based on tables available in MPEG-4 core experiment T5 and are available publicly.

Adaptive Deinterleaved Transform Coding

Further improvements in DT coding are possible by using an encoding structure as shown in FIG 16, which shows the block diagram of an Adaptive Global Deinterleaved Transform (AGDT) encoder 160. The major difference with respect to FIG 1 is that a quadtree segmentation is employed by using a Quadtree Segmenter on an entire picture or VOP basis prior to the deinterleaver and is adaptive rather than fixed segmentation. Thus, the deinterleaving is only performed on only the portions identified by Global Quadtree Segmenter to be worth deinterleaving while others are coded without deinterleaving. The operation of other blocks is similar to that discussed for fixed GDT.

Referring to FIG 16, the image is fed into the Global Quadtree Segmenter 161, whose output is passed to the deinterleaver 162, which in turn passes its output to the transform 163. The QuantX 164 receives the output from the transform 163 and passes is output to the entropy encoder 165, which outputs the coded bitstream.

FIG 17 shows the block diagram of the AGDT decoder 170 corresponding to the AGDT encoder 160 shown in FIG 16. The coded bitstream is fed into the entropy decoder 171, the output of which is passed to the inverse QuantX 172, which in tum passes its output to the inverse transform 173. The reinterleaver 174 receives the output from the inverse transform 173 and feeds its output to the Global Quadtree Assembler 175, which outputs the reconstructed image.

FIG 18 shows a block diagram of an Adaptive Local Deinterleaved Transform (ALDT) encoder 180. The major difference with respect to FIG 16 is that the quadtree segmentation is applied locally (on regions) rather than the entire picture or VOP. Deinterleaving is then performed on the region identified by the Local Quadtree Segmenter as worth deinterleaving. The remaining blocks are similar to those described above.

The image signal is input to the Local Quadtree Segmenter 181, whose output is fed to the deinterleaver 182, which passes its output to the transform 183. The QuantX 184 receives the output from the transform 183 and passes its output to the entropy encoder 185, which outputs the coded bitstream.

FIG 19 shows the ALDT decoder 190 that corresponds to the encoder of FIG 18. The coded bits are fed into the entropy decoder 191, which passes its output to the inverse QuantX 192, which in turn passes its output to the inverse transform 193, which in turn passes its output to the reinterleaver 194. The local quadtree assembler 195 receives the output of the reinterleaver 194 and outputs the reconstructed image.

Quadtree Segmenter

As shown in FIGs 16 and 18, quadtree segmentation is employed prior to deinterleaving to allow adaptation of the amount of deinterleaving to the spatial content of the picture being coded.

An example of quadtree segmentation employed is shown in FIG 20; both GDT and LDT use this type of segmentation, the only difference being in the number of levels employed -- GDT employs _ levels of segmentation, whereas LDT employs _ levels of segmentation.

As shown in FIG 20, picture block 200 is segmented into subblocks 202-205. Subblock 203 is then further partitioned into sections 206-209. The remaining blocks were not segmented to indicate that this process only segments the necessary blocks.

Syntax and Semantics for MPEG-4

We now provide the necessary syntax and semantics needed for generating coded bitstreams using the present invention. The various classes referred to below correspond to the current syntax of MPEG-4 VM3.2

VideoSession Class

No changes are necessary for this class.

VideoObject Class

No changes are necessary for this class.

VideoObject Layer Class

No changes are necessary for this class.

VideoObject Plane Class

Two new syntax elements are introduced in this

class as follows.

:
:
region_size
deinterleave_ratio
:
:

These syntax elements are defined as follows.

region_size

This is a 3 bit code which specifies the size of the region on which deinterleaving is performed prior to coding. The size of the region for each code is shown in Table 1 as follows:

Table 1

Code	Meaning
000	16x16
001	32x32
010	64x64
011	128x128
100	reserved
101	reserved
110	reserved
111	full picture

deinterleave_ratio

This is the 3-bit code which specifies the amount of deinterleaving performed on the identified region prior to coding. The same deinterleaving ration is used both horizontally and vertically. The amount of deinterleaving for each code is shown in Table 2 as follows.

Table 2

Code	Meaning
000	1:1
001	2:1
010	4:1
011	8:1
100	16:1
101	reserved
110	reserved
111	reserved

Region Class

Data for each region consists of region header followed by subregion data.

Table 3

Rtype	Rquant	Subregion data

Rquant

Rquant is a 3 bit quantizer that takes nonlinear values bounded by 1 to 31 with the meaning as shown in Table 4.

Table 4

Code	Qp
000	2
001	43
010	7
011	10
100	14
101	18
110	23
111	28

Subregion Class

The definition of subregion data is dependent on the QuantX method employed and is specified as follows:
For QuantX Method 1:

Table 5

Structure of Subregion Class for QuantX Method 1	
Cod_subreg	Tcoefs_subreg

Cod_subreg

Cod-subreg is a 1 bit flag that identifies if there is any coded data (nonzero values) for that subregion.

Tcoefs_subreg

Tcoefs_subreg are differential quantized coefficients of subregion.
For QuantX Method 2:

Table 6

Structure of Subregion Class for QuantX Method 2	
Cod_vector	Tcoefs_vector

Cod_vector

Cod-vector is a 1 bit flag that identifies if there is any coded data a subregion.

Tcoefs_vector

Tcoefs_vector refers to twice quantized coefficients of a vector.
For QuantX Method 3:

Claims

1. A method for coding an image comprising the steps of:

   a) deinterleaving the image to form a plurality of image subsets;
   b) transforming the plurality of image subsets into a plurality of transform coefficients;
   c) converting the plurality of transform coefficients to a plurality of samples; and
   d) performing an entropy encoding of the plurality of samples to form an encoded bit stream.

2. A method for decoding a bit stream representing an image that has been encoded, comprising the steps of:

   a) performing an entropy decoding of the bit stream to form a plurality of samples;
   b) converting the plurality of samples to a plurality of transform coefficients;
   c) performing an inverse transformation on the plurality of transform coefficients to form a plurality of image subsets; and
   d) reinterleaving the plurality of image subsets to form the image.

3. The method according to claim 2, wherein the step b) of converting comprises the steps of:

   (i) performing an inverse scanning of the plurality of samples to form a plurality of difference values;
   (ii) adding the plurality of difference values to a plurality of predicted values to form a plurality of quantized values;
   (iii) performing a DC and AC coefficient prediction on the plurality of quantized values to form the plurality of predicted values; and
   (iv) performing an inverse quantization of the plurality of quantized values to form the plurality of transform coefficients.

4. The method according to claim 2, wherein the step b) of converting comprises the steps of:

   (i) performing an inverse scan of the plurality of samples to form a plurality of quantized values;
   (ii) performing an inverse quantization of the plurality of quantized values to form a plurality of transformed vectors using a first quantization level;
   (iii) performing an inverse transformation of the plurality of transformed vectors to form a plurality of vectors;
   (iv) generating a plurality of quantized transform coefficients from the plurality of vectors; and
   (v) performing an inverse quantization of the plurality of quantized transform coefficients to form the plurality of transform coefficients using a second quantization level, wherein said first quantization level is greater than said second quantization level.

5. The method according to claim 2, wherein the step b) of converting comprises the steps of:

   (i) performing a vector quantization indices reordering of the plurality of samples to form a plurality of quantized vectors;
   (ii) performing an inverse vector quantization of the plurality of quantized vectors to form a plurality of limited dimension vectors;
   (iii) normalizing a dimension of the plurality of limited dimension vectors to form a plurality of vectors; and
   (iv) unformatting the plurality of vectors to form the plurality of transform coefficients.

6. The method according to claim 2, wherein the step d) of reinterleaving further comprises the steps of:

   (i) reinterleaving the plurality of image subsets to form a plurality of segments; and
   (ii) assembling the plurality of segments formed by the reinterleaving step d) (i) into the image.

7. A method for coding an image comprising the steps of:

   a) segmenting the image into a plurality of local regions;
   b) deinterleaving the plurality of local regions to form a plurality of deinterleaved regions;
   c) transforming the plurality of deinterleaved regions into a plurality of transform coefficients;
   d) converting the plurality of transform coefficients to a plurality of samples; and
   e) performing an entropy encoding of the plurality of samples to form an encoded bit stream.

8. A method for decoding a bit stream representing an image that has been encoded, comprising the steps of:

   a) performing an entropy decoding of the bit stream to form a plurality of samples;

b) converting the plurality of samples to a plurality of transform coefficients;
c) performing an inverse transformation of the plurality of transform coefficients to form a plurality of deinterleaved regions;
d) reinterleaving the plurality of deinterleaved regions to form a plurality of local regions; and
e) assembling the plurality of local regions to form the image.

9. The method according to claim 8, wherein the step b) of converting comprises the steps of:

(i) performing an inverse scanning of the plurality of samples to form a plurality of difference values;
(ii) adding the plurality of difference values to a plurality of predicted values to form a plurality of quantized values;
(iii) performing a DC and AC coefficient prediction on the plurality of quantized values to form the plurality of predicted values; and
(iv) performing an inverse quantization of the plurality of quantized values to form the plurality of transform coefficients.

10. The method according to claim 8, wherein the step b) of converting comprises the steps of:

(i) performing an inverse scan of the plurality of samples to form a plurality of quantized values;
(ii) performing an inverse quantization of the plurality of quantized values to form a plurality of transformed vectors using a first quantization level;
(iii) performing an inverse transformation of the plurality of transformed vectors to form a plurality of vectors;
(iv) generating a plurality of quantized transform coefficients from the plurality of vectors; and
(v) performing an inverse quantization of the plurality of quantized transform coefficients to form the plurality of transform coefficients using a second quantization level, wherein said first quantization level is greater than said second quantization level.

11. The method according to claim 8, wherein the step b) of converting comprises the steps of:

(i) performing a vector quantization indices reordering of the plurality of samples to form a plurality of quantized vectors;
(ii) performing an inverse vector quantization of the plurality of quantized vectors to form a plurality of limited dimension vectors;
(iii) normalizing a dimension of the plurality of limited dimension vectors to form a plurality of vectors; and
(iv) unformatting the plurality of vectors to form the plurality of transform coefficients.

12. The method according to claim 8, wherein the step d) of reinterleaving further comprises the steps of:

(i) reinterleaving the plurality of deinterleaved regions to form a plurality of local region segments; and
(ii) assembling the plurality of local region segments to form the plurality of local regions.

13. An apparatus for coding an image comprising:

a) a deinterleaver receiving the image and deinterleaving the image to form a plurality of image subsets;
b) a transformer being coupled to the deinterleaver and transforming the plurality of image subsets into a plurality of transform coefficients;
c) a converter being coupled to the transformer and converting the plurality of transform coefficients to a plurality of samples; and
d) an encoder being coupled to the converter and performing an entropy encoding of the plurality of samples to form an encoded bit stream.

14. An apparatus for decoding a bit stream representing an image that has been encoded, comprising:

a) a decoder receiving the bit stream and performing an entropy decoding of the bit stream to form a plurality of samples;
b) a converter being coupled to the decoder and converting the plurality of samples to a plurality of transform coefficients;
c) a reverse transformer being coupled to the converter and performing an inverse transformation of the plurality of transform coefficients to form a plurality of image subsets; and
d) a reinterleaver being coupled to the reverse transformer and reinterleaving the plurality of image subsets to form the image.

15. The apparatus according to claim 14, wherein the converter comprises an inverse extended quantizer, said inverse extended quantizer including:

a) an inverse scanner being coupled to the decoder and performing an inverse scan on the plurality of samples to form a plurality of difference values;
b) an adder having a first input being coupled to the inverse scanner, having a second input, and having an output outputting a plurality of quantized values;

c) a coefficient predictor having an input being coupled to the output of the adder, performing a DC and AC coefficient prediction on the plurality of quantized values to form a plurality of predicted values, and having an output being coupled to the second input of the adder, wherein the plurality of quantized values equals a sum of the plurality of difference values and the plurality of predicted values; and
d) an inverse quantizer being coupled to the adder and inverse quantizing the plurality of quantized values to form the plurality of transform coefficients.

16. The apparatus according to claim 14, wherein the converter comprises an inverse extended quantizer, said inverse extended quantizer including:

a) an inverse scanner being coupled to the decoder and performing an inverse scan of the plurality of samples to form a plurality of quantized values;
b) a first inverse quantizer being coupled to the inverse scanner and performing an inverse quantization of the plurality of quantized values to form a plurality of transformed vectors using a first quantization level;
c) an inverse transformer being coupled to the inverse quantizer and performing an inverse transformation of the plurality of transformed vectors to form a plurality of vectors;
d) a vector unformatter being coupled to the inverse transformer and generating a plurality of quantized transform coefficients from the plurality of vectors; and
e) a second inverse quantizer being coupled to the vector unformatter and generating the plurality of transform coefficients from the plurality of quantized transform coefficients using a second quantization level, wherein the first quantization level is greater than the second quantization level.

17. The apparatus according to claim 14, wherein the converter comprises an inverse extended quantizer, said inverse extended quantizer comprising:

a) a vector quantization indices reorderer being coupled to the decoder and performing a vector quantization indices reordering of the plurality of samples to form a plurality of quantized vectors;
b) an inverse vector quantization being coupled to the vector quantization indices reorder and performing an inverse vector quantization of the plurality of quantized vectors to form a plurality of limited dimension vectors;
c) a dimension normalizer being coupled to the inverse vector quantization and normalizing a dimension of the plurality of limited dimension vectors to form a plurality of vectors; and
d) a vector unformatter being coupled to the dimension normalizer and unformatting the plurality of vectors to form the plurality of transform coefficients.

18. The apparatus according to claim 14, further comprising:

a) a global quadtree assembler being coupled to the reinterleaver, wherein the reinterleaver reinterleaves the plurality of image subsets to form a plurality of segments, and the global quadtree assembler assembles the plurality of segments formed by the reinterleaver into the image.

19. A system for coding an image comprising:

a) a segmenter receiving the image and segmenting the image into a plurality of local regions;
b) a deinterleaver being coupled to the segmenter and deinterleaving the plurality of local regions to form a plurality of local region subsets;
c) a transformer being coupled to the deinterleaver and transforming the plurality of local region subsets into a plurality of transform coefficients;
d) a converter being coupled to the transformer and converting the plurality of transform coefficients to a plurality of samples; and
e) an encoder being coupled to the converter and performing an entropy encoding of the plurality of samples to form an encoded bit stream.

20. A system for decoding a bit stream representing an image that has been encoded, comprising:

a) a decoder receiving the bit stream and performing an entropy decoding of the bit stream to form a plurality of samples;
b) a converter being coupled to the decoder and converting the plurality of samples to a plurality of transform coefficients;
c) a reverse transformer being coupled to the converter and performing an inverse transformation of the plurality of transform coefficients to form a plurality of local region subsets;
d) a reinterleaver being coupled to the reverse transformer and reinterleaving the plurality of local region subsets to form a plurality of local regions; and
e) an assembler being coupled to the reinterleaver and assembling the plurality of local re-

gions to form the image.

FIG. 1

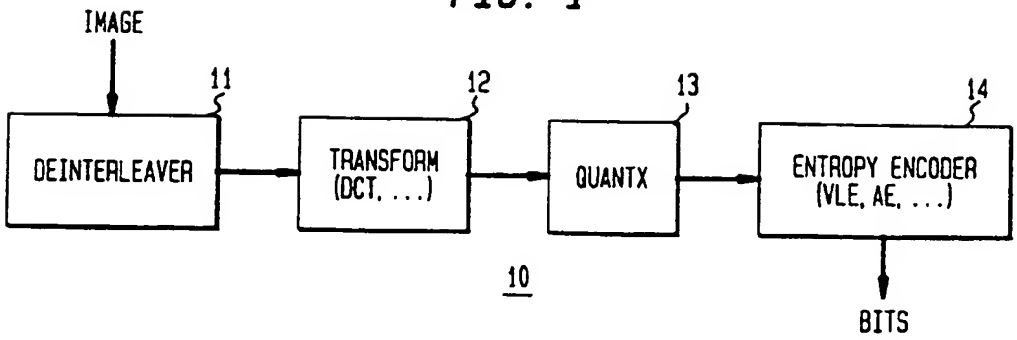


FIG. 2

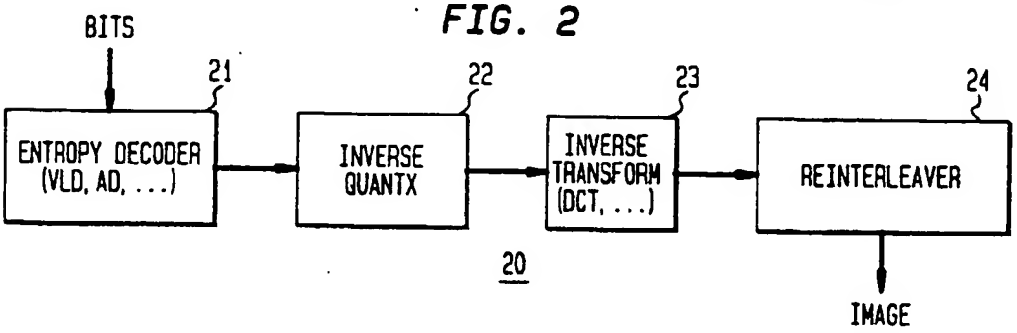


FIG. 3

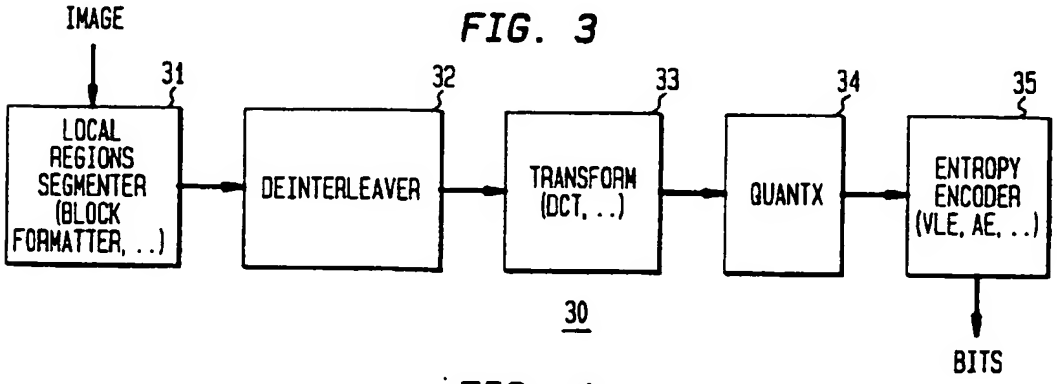


FIG. 4

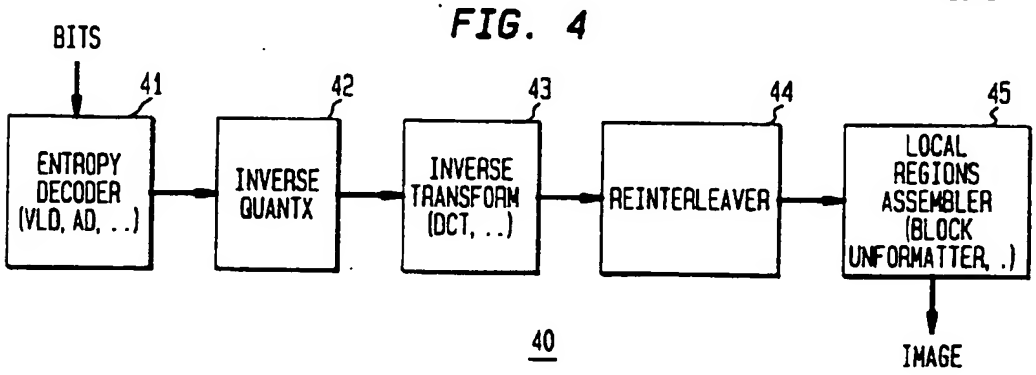


FIG. 5

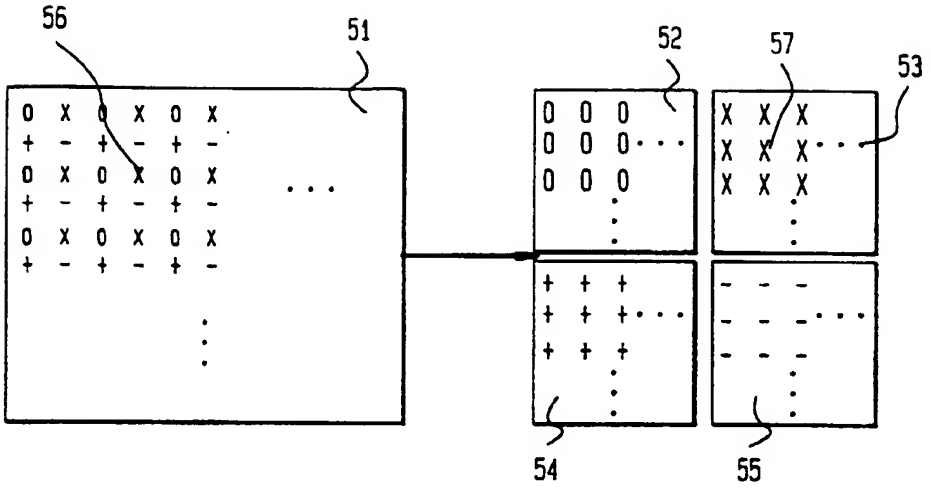


FIG. 6

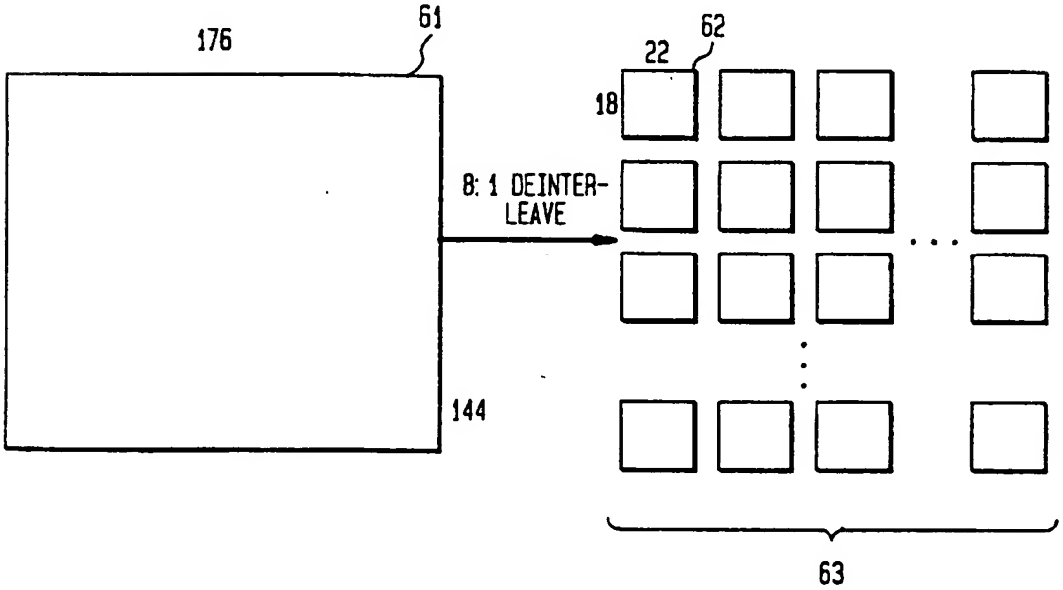


FIG. 7

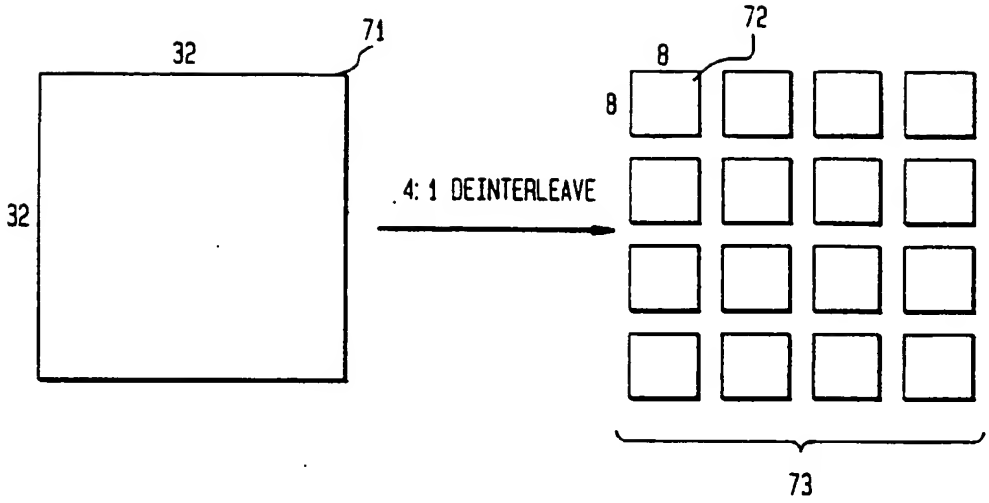
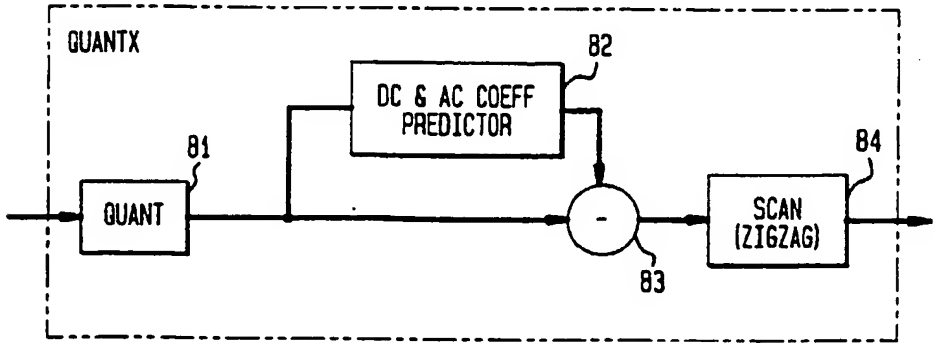

32
71
32
4:1 DEINTERLEAVE
72
8
8
73

FIG. 8


QUANTX
82
DC & AC COEFF PREDICTOR
81
84
QUANT
-
SCAN (ZIGZAG)
83

80

FIG. 9

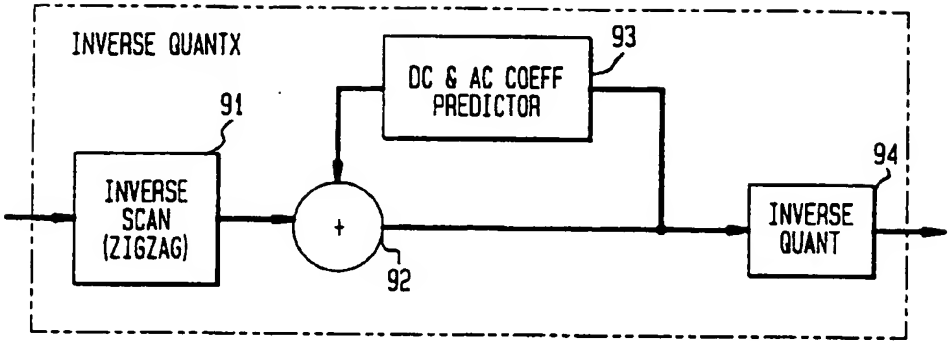

INVERSE QUANTX
91
INVERSE SCAN (ZIGZAG)
+
92
DC & AC COEFF PREDICTOR
93
INVERSE QUANT
94
90

FIG. 10

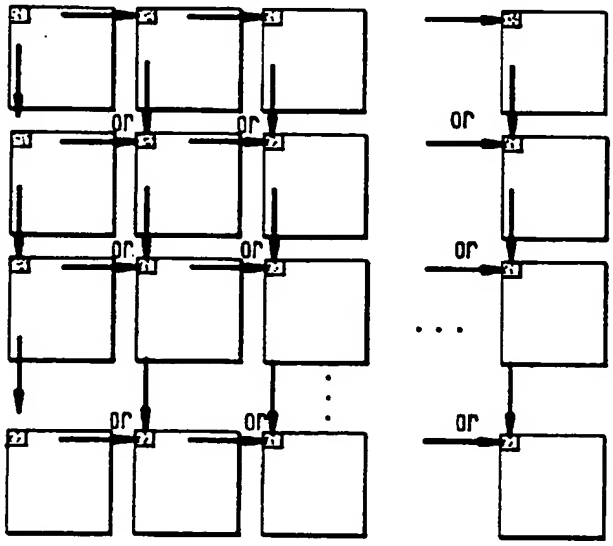

or

FIG. 11

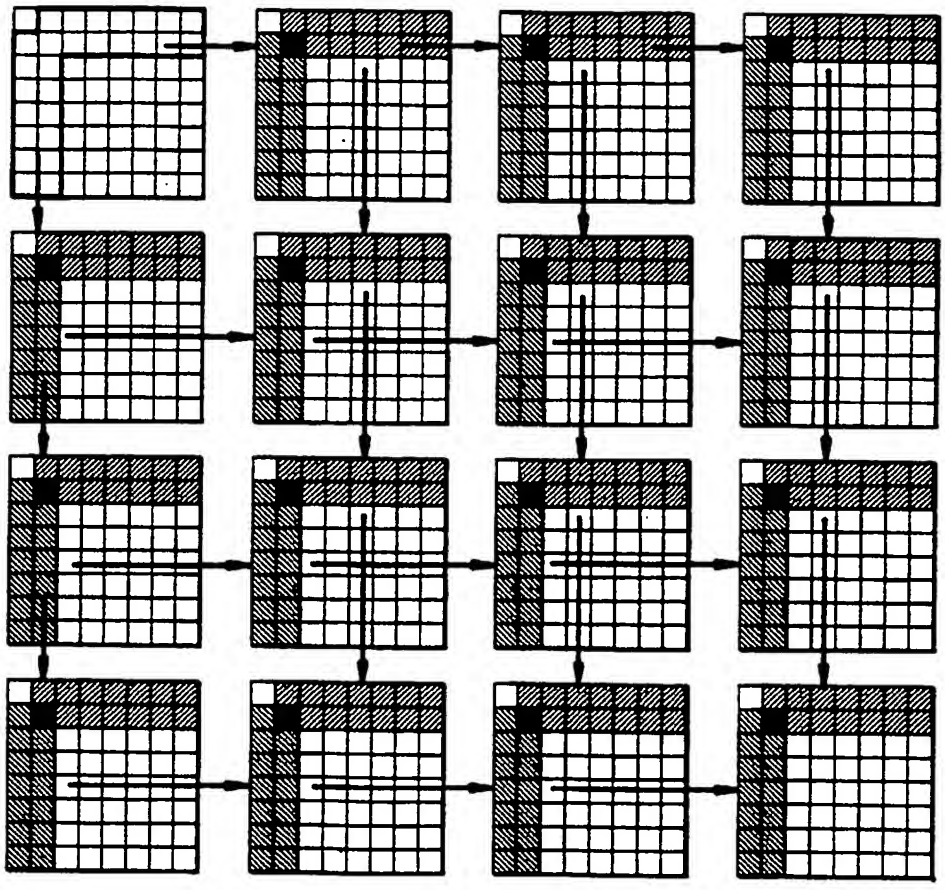


FIG. 12

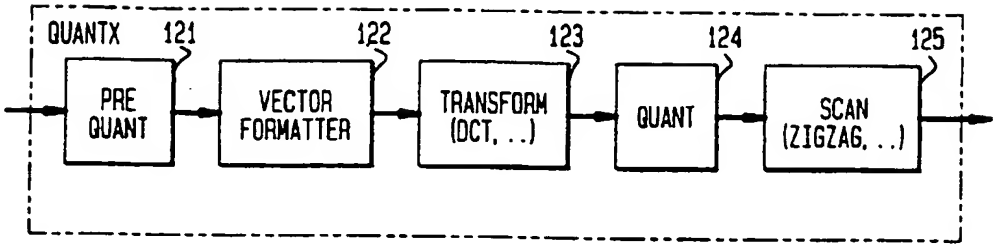


FIG. 13

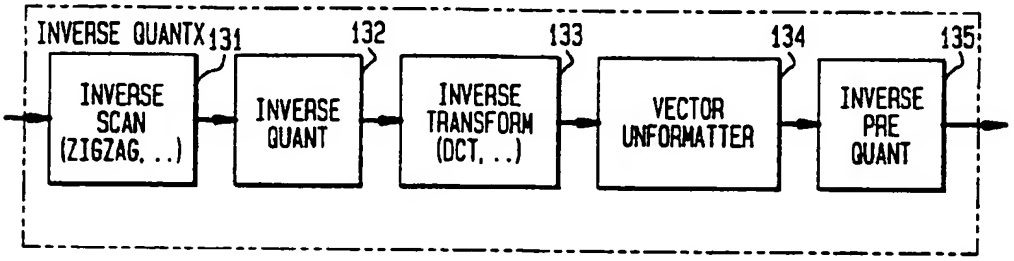


FIG. 14

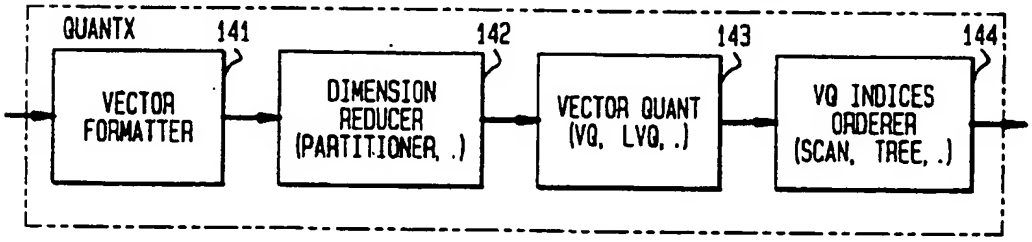


FIG. 15

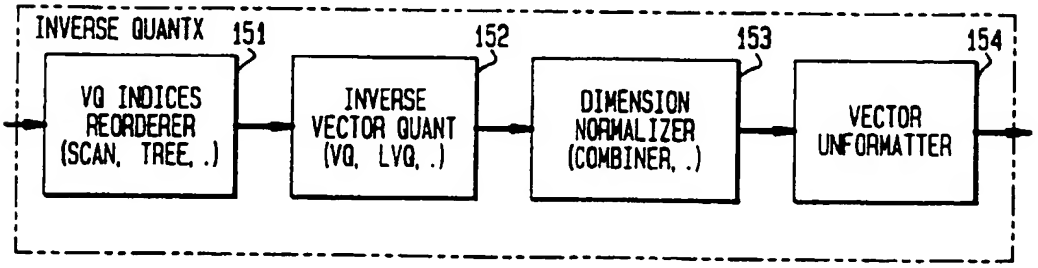


FIG. 16

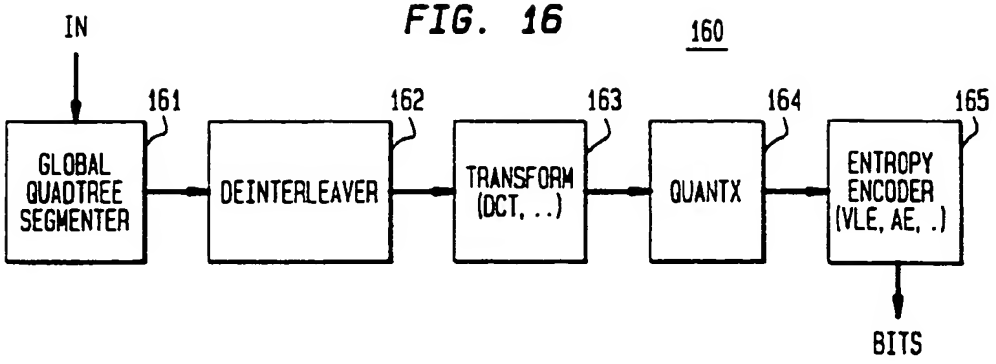


FIG. 17

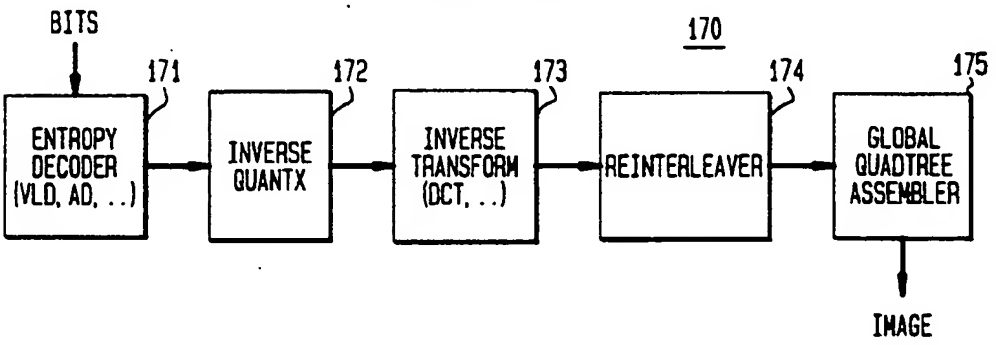


FIG. 18

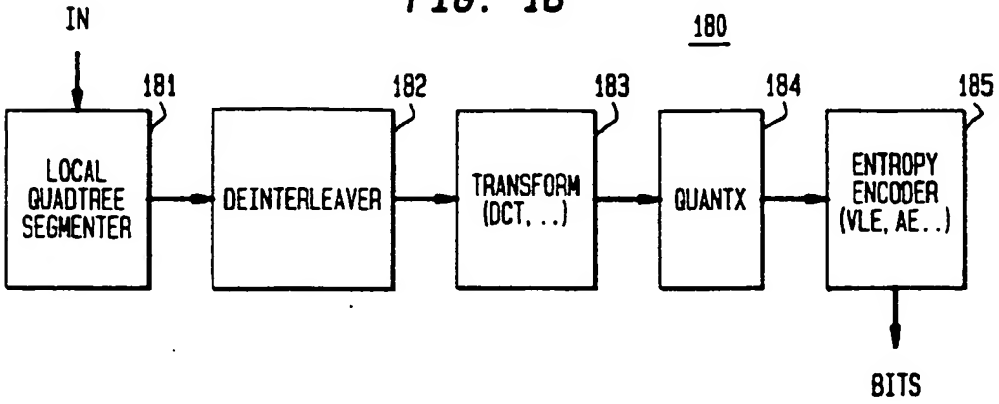


FIG. 19

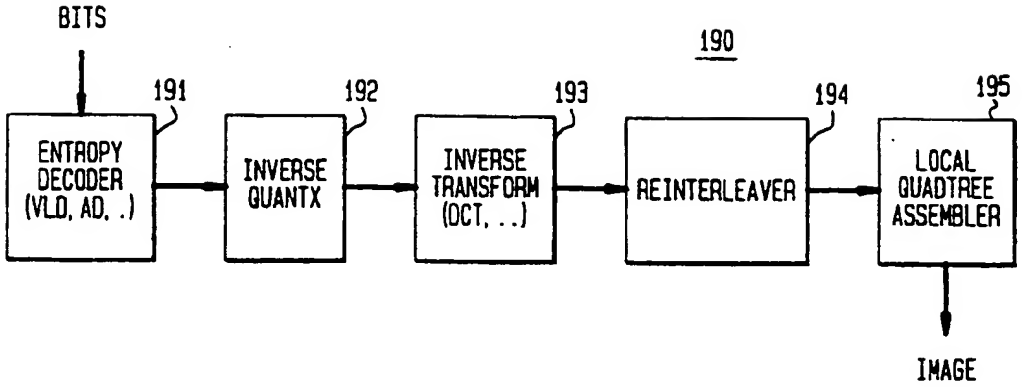
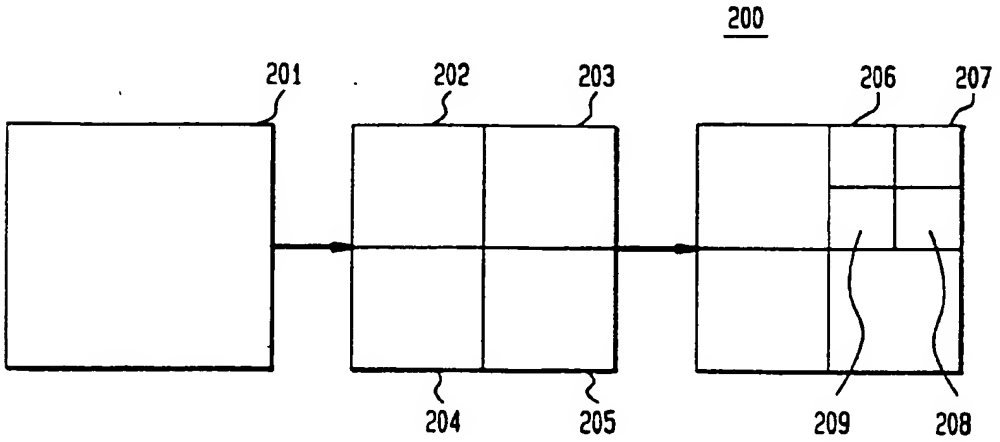

BITS
190
191
192
193
194
195
ENTROPY DECODER (VLD, AD, .)
INVERSE QUANTX
INVERSE TRANSFORM (DCT, ..)
REINTERLEAVER
LOCAL QUADTREE ASSEMBLER
IMAGE

FIG. 20


200
201
202
203
206
207
204
205
209
208